Omnet/PhoenixSim Component
Author: Rob Hendry (robert.i.hendry@gmail.com)

This component includes the complete Omnet++ v4.1 network simulator command-line environment as well as PhoenixSim, a library built on top of Omnet containing libraries for running simulations of photonic networks.

To Install:

1. Download and install Omnet++ 4.1 (http://www.omnetpp.org/omnetpp/cat_view/17-downloads/1-omnet-releases).

2. Run the patch located in the PhoenixSim/patch directory, passing the omnetpp-4.1/include directory full path (or some directory above it) as an argument. This simply adds the “virtual” qualifier to the cSimulation::insertMsg() function in its declaration in <omnetpp_dir>/include/csimulation.h.

3. Make and install the PhoenixSim component.

To Use:

When instantiating a PhoenixSim component in sdl.xml, it requires a few things:

1. There MUST be a link defined with the name “slink” and zero latency. This will be the self-link that the component uses to expose Omnet messages to SST as they reach the front of the event queue.

2. There MUST be the following component parameters:

<ini>ini_file_location</ini>: This specifies where the top-level parameter file is located for the Omnet simulation.

<timebase>SST_timebase</timebase>: This is an SST-format time base which defines the SST component's time base as well as the Omnet simulation granularity (they must match).

<nedsource>top_ned_directory</nedsource>: This should point to the top level folder of your Omnet simulation. It directs Omnet where to start looking for .ned files. Note that Omnet packages are defined relative to this directory.

<network>top_level_network_name</network>: This should be the full name (package names included) of the top level network that defines the network model for your Omnet simulation.

<configuration>config_name</configuraiton>: This is simply the name of the configuration you wish to run.

<run>run_number</run>: This is the run number you wish to run.

There are a few optional component parameters as well:

<mode>[normal, fast, express]</mode>: Normal and fast modes will print out more information about messages in the simulation (right now, normal and fast modes are identical). Express mode only prints info periodically, and greatly reduces the run time of the simulation.

<v></v>, <h>arg</h>, <a></a>, <x>arg,</x>, <g></g>, <G></G>: These substitute for the -v, -h, -a, -x, -g, and -G command line Omnet options.

Setting an Omnet Simulation:

Create a subdirectory with the elements/PhoenixSim directory. All of your models, network descriptions, and parameter files (i.e. .h, .cc, .ned, and .ini files– everything that you define for your simulation) need to be within a subdirectory of elements/PhoenixSim. You can use Omnet++'s graphical IDE to develop your Omnet simulation, but you will most likely have to change some package names and possibly other things when you decide to integrate with SST.

Editing Makefile.am: You will have to add the paths of all .h and .cc files included in your Omnet simulation to Makefile.am and then rebuild the entire PhoenixSim component. For convenience, there are two variables defined near the top of the make file: PROJECT_SOURCES, and PROJECT_PATHS. You can simply list all of your .h and .cc files after PROJECT_SOURCES to include these files in the build. You should also define all the necessary include paths for your simulation by listing them after PROJECT_PATHS with the -I before each.

Note: If you use the Omnet++ graphical IDE to develop your simulation, you can easily look in the Makefile automatically generated by Omnet to get a list of your model's files, and a list of all the include paths. These can be copied over to Makefile.am, with the caveat that (I think) the Omnet-generated Makefile lists .o files, instead of the individual .h and .cc files.

Using PhoenixSim:

PhoenixSim is automatically included in the component build. However, in order to use PhoenixSim modules, you must develop your simulation within the elements/PheonixSim/PhoenixSim directory, and set the nedsource option to be “PhoenixSim.” The sample.xml file has a configuration set up to run an example PhoenixSim simulation. Notice the network name starts with packages that correspond to subdirectories of elements/PhoenixSim/PhoenixSim.